Econ 603 - A2 - Javier Fernandez

Assignment #2 ; Class: Econ 613 ; Name: Javier Fernandez ; Date: March 24th, 2021 ;

Preliminaries ——

```
# Libraries
library(tidyverse)
library(bayesm)
library(nloptr)
library(mlogit)

# Set WD
setwd("C:/Users/javie/OneDrive/Documents/GitHub/ECON613/Assingments/A2")
```

——– 1 - Data Description ——-

Invoke dataset

```
data("margarine")
```

a. Average and dispersion of prices

```
# By product
avg_price <- apply(margarine$choicePrice[,3:12], 2, mean)
sd_price <- apply(margarine$choicePrice[,3:12], 2, sd)
price_table <- t(rbind(avg_price,sd_price))
price_table
```

```
##          avg_price   sd_price
## PPk_Stk  0.5184362 0.15051740
## PBB_Stk  0.5432103 0.12033186
## PFl_Stk  1.0150201 0.04289519
## PHse_Stk 0.4371477 0.11883123
## PGen_Stk 0.3452819 0.03516605
## PImp_Stk 0.7807785 0.11464607
## PSS_Tub  0.8250895 0.06121159
## PPk_Tub  1.0774094 0.02972613
## PFl_Tub  1.1893758 0.01405451
## PHse_Tub 0.5686734 0.07245500
```

b. Market share (choice frequency) and market share by product characteristics (choice frequency by price bins: below average, over average)

Market share

```
mkt_share <- 100*table(margarine$choicePrice[,2])/nrow(margarine$choicePrice)
names(mkt_share) <- names(margarine$choicePrice[,3:12])
print(mkt_share)
```

```
##    PPk_Stk    PBB_Stk    PFl_Stk   PHse_Stk   PGen_Stk   PImp_Stk    PSS_Tub    PPk_Tub
## 39.507830 15.637584  5.436242 13.266219  7.046980  1.655481  7.136465  4.541387
##    PFl_Tub   PHse_Tub
##  5.033557  0.738255
```

Market share by product characteristics

```
# Generate a dummy variable indicating if the product was bought below or above price
avg_price <- apply(margarine$choicePrice[,3:12], 2, mean)
avg_price <- data.frame("product"=names(avg_price),"avg_price"=avg_price)

    temp=NULL
    chose=NULL
    price=NULL
    avg_price_of_choice=NULL

  for(i in 1:nrow(margarine$choicePrice)){
    chose[i]=margarine$choicePrice$choice[i]
    price[i]=margarine$choicePrice[i,chose[i]+2]
    avg_price_of_choice[i]=avg_price[chose[i],2]
    temp=rbind(temp,c(chose[i],price[i],avg_price_of_choice[i]))
  }
  colnames(temp) <- c("Choice","Price","Avg_Price")
  temp <- temp %>% as.data.frame() %>%
  mutate(below_above_avg=ifelse(Price>Avg_Price,"Above","Below"))

# By price bins
table(temp$Choice,temp$below_above_avg) %>% prop.table(.,margin = 2)
```

```
##
##             Above       Below
##    1  0.376853180 0.411097100
##    2  0.125777140 0.183270282
##    3  0.025346724 0.079865490
##    4  0.141559063 0.124842371
##    5  0.066953611 0.073560319
##    6  0.008608321 0.023539302
##    7  0.095648015 0.050021017
##    8  0.055475849 0.036569987
##    9  0.095648015 0.010508617
##    10 0.008130081 0.006725515
```

c. Illustrate the mapping between observed attributes and choices. (Which customers are choosing which products?)

```
merged_data <- left_join(margarine$choicePrice,margarine$demos)
merged_data <- merged_data %>% mutate(Income_bins=case_when(
              Income<20~"< 20",
              Income>=20 & Income<40 ~ "20-39",
              Income>=40 & Income<60 ~ "40-59",
              Income>=60~"> 60"
))
merged_data$Income_bins <- factor(merged_data$Income_bins, levels = c("< 20","20-39","40-59","> 60"))
choice_by_incomebins <- 100*prop.table(table(merged_data$choice,merged_data$Income_bins),2)
print(choice_by_incomebins)
```

```
##
##            < 20      20-39      40-59       > 60
##   1  42.8477258 38.5654401 36.8497110 28.9473684
##   2  17.4027686 15.6031672 12.2832370 13.1578947
##   3   5.3394858  4.0987424  9.6820809  6.1403509
##   4  12.5906394 14.6250582 10.2601156 14.9122807
##   5   4.5484509 10.1537028  2.8901734  7.0175439
##   6   1.0547132  0.6054960  5.7803468  4.3859649
##   7   9.0309822  6.0083838  6.5028902  7.0175439
##   8   2.2412657  4.3782021 10.4046243  2.6315789
##   9   4.5484509  4.9371216  4.7687861 14.9122807
##   10  0.3955175  1.0246856  0.5780347  0.8771930
```

Fam size

```
100*prop.table(table(merged_data$choice,merged_data$Fam_Size),1)
```

```
##
##                1           2           3           4           5           6
##   1   8.38052095 26.84031710 22.65005663 28.42582106  9.06002265  4.30351076
##   2   7.01001431 30.32904149 23.60515021 27.89699571  7.58226037  3.14735336
##   3  15.63786008 50.61728395 11.93415638 13.58024691  8.23045267  0.00000000
##   4   3.87858347 25.96964587 20.06745363 30.18549747 12.14165261  5.56492411
##   5   3.17460317 17.46031746 19.04761905 40.31746032 10.47619048  7.61904762
##   6   9.45945946 35.13513514 14.86486486  9.45945946 31.08108108  0.00000000
##   7   7.83699060 36.67711599 24.13793103 25.07836991  2.50783699  3.76175549
##   8   8.86699507 25.61576355 22.66009852 37.43842365  0.98522167  4.43349754
##   9  15.11111111 49.77777778 21.33333333  8.88888889  4.88888889  0.00000000
##   10  0.00000000  9.09090909  9.09090909 27.27272727 39.39393939 15.15151515
##
##                7           8
##   1   0.05662514  0.28312571
##   2   0.14306152  0.28612303
##   3   0.00000000  0.00000000
##   4   1.34907251  0.84317032
##   5   0.63492063  1.26984127
```

```
## 6 0.00000000 0.00000000
## 7 0.00000000 0.00000000
## 8 0.00000000 0.00000000
## 9 0.00000000 0.00000000
## 10 0.00000000 0.00000000
```

College

```
100*prop.table(table(merged_data$choice,merged_data$college),1)
```

```
##
##            0        1
##   1  68.23330 31.76670
##   2  68.66953 31.33047
##   3  54.73251 45.26749
##   4  70.65767 29.34233
##   5  72.69841 27.30159
##   6  56.75676 43.24324
##   7  67.71160 32.28840
##   8  74.38424 25.61576
##   9  72.44444 27.55556
##   10 54.54545 45.45455
```

Retired

```
100*prop.table(table(merged_data$choice,merged_data$retired),1)
```

```
##
##             0         1
##   1  80.067950 19.932050
##   2  75.965665 24.034335
##   3  46.913580 53.086420
##   4  84.654300 15.345700
##   5  85.396825 14.603175
##   6  62.162162 37.837838
##   7  85.266458 14.733542
##   8  90.147783  9.852217
##   9  64.000000 36.000000
##   10 87.878788 12.121212
```

White collar

```
100*prop.table(table(merged_data$choice,merged_data$whtcollar),1)
```

```
##
##             0         1
##   1  42.978482 57.021518
```

```
## 2 45.636624 54.363376
## 3 45.679012 54.320988
## 4 40.809444 59.190556
## 5 28.571429 71.428571
## 6 43.243243 56.756757
## 7 42.319749 57.680251
## 8 42.857143 57.142857
## 9 42.222222 57.777778
## 10 6.060606 93.939394
```

——— 2 - First Model ———-

The model proposed is a conditional logit of the form:

$$Demand_{ij} = \alpha_j + \beta * price_{ij}$$

Conditional Logit Function

```
# Conditional Logit Function
conditional_logit = function(parameters,choice,x1){
  ni = nrow(x1)
  nj = 10
  ut = mat.or.vec(ni,nj)

  ut[,1] = parameters[10]*x1[,1] # intercept=0
  for (j in 2:nj)
  {
    # conditional logit
    ut[,j] = parameters[j-1] + parameters[10]*x1[,j]
  }
  prob   = exp(ut)
  prob   = sweep(prob,MARGIN=1,FUN="/",STATS=rowSums(prob))

  # Probabilities times Indicator of choice
  probc = NULL
  for (i in 1:ni)
  {
    probc[i] = prob[i,choice[i]]
  }
  probc[probc>0.999999] = 0.999999
  probc[probc<0.000001] = 0.000001
  like = sum(log(probc))
  return(-like)
}
```

Estimating First Model

```
# Initial values for optimization
npar=10
lower  = rep(-10,npar)
upper  = rep(10,npar)
```

```r
start  = runif(npar)

# Variables
choice=margarine$choicePrice[,2]
x1=margarine$choicePrice[,3:12]

# Optimizing to get the results
res_cond_logit = optim(start,fn=conditional_logit,method="BFGS",
                       control=list(trace=6,REPORT=20,maxit=10000),choice=choice,x1=x1,)
```

```
## initial  value 11598.668272
## final  value 7464.932069
## converged
```

Result:

The coefficient on price is negative indicating that an increase in the price of the option j reduces the demand of option j.

——— 3 - Second Model ———-

The model proposed is a multinomial logit of the form:

$$Demand_{ij} = \alpha_j + \beta_j^1 * Income_i + \beta_j^2 * Fam\_Size_i + \beta_j^3 * College_i + \beta_j^4 * Whtcollar_i + \beta_j^5 * Retired_i$$

Multinomial Logit Function

```r
# Multinomial Logit Function
multinomial_logit = function(param,choice,X,n_alternatives){
  #Coefficients and preliminaries
  ni = nrow(X)
  nj = n_alternatives
  intercepts = c(0,param[1:(nj- 1)])
  slopes1 = c(0,param[nj:(2*(nj-1))])
  slopes2 = c(0,param[(2*(nj-1)+1):(3*(nj-1))])
  slopes3 = c(0,param[(3*(nj-1)+1):(4*(nj-1))])
  slopes4 = c(0,param[(4*(nj-1)+1):(5*(nj-1))])
  slopes5 = c(0,param[(5*(nj-1)+1):length(param)])
  ut = mat.or.vec(ni,nj)

  #Variables
  Income = X[,1]
  Fam_size = X[,2]
  college = X[,3]
  whtcollar = X[,4]
  retired = X[,5]

  # Loop to compute probabilites
  ut[,1] = 0 # intercept =0 and slopes=0
  for (j in 2:nj){
    # multinomial logit
```

```r
    ut[,j] = intercepts[j] + slopes1[j]*Income + slopes2[j]*Fam_size +
      slopes3[j]*college + slopes4[j]*whtcollar + slopes5[j]*retired
  }
  prob   = exp(ut)
  prob   = sweep(prob,MARGIN=1,FUN="/",STATS=rowSums(prob))
  probc = NULL
  for (i in 1:ni)
  {
    probc[i] = prob[i,choice[i]]
    if(is.na(probc[i])){
      break
      }
  }
  probc[probc>0.999999] = 0.999999
  probc[probc<0.000001] = 0.000001
  like = sum(log(probc))
  return(-like)
}
```

Estimating Second Model

```r
# Data for the estimation
X <- merged_data %>% select(Income,Fam_Size,college,whtcollar,retired)

# Initial values for model estimation
set.seed(12)
n_alternatives=10
npar=(n_alternatives-1)*6
lower  = rep(-10,npar)
upper  = rep(10,npar)
start  = runif(npar)

# Optimization
res_multlogit=optim(start,fn=multinomial_logit,method="BFGS",control=list(trace=6,REPORT=20,maxit=10000)
```

```
## initial  value 46226.285255
## iter  20 value 11017.572628
## iter  40 value 8856.261417
## iter  60 value 8529.512301
## iter  80 value 8284.153355
## iter 100 value 8020.457522
## iter 120 value 7932.889101
## final  value 7931.686143
## converged
```

Result:

The coefficients on family income is negative for options 2, 5, 7, and 10 indicate that as family income increase the probability of demanding this option diminishes in comparison to option 1. for options 3, 4, 6, 8, and 9 as family income increase the probability of demanding this option increases in comparison to option 1.

——— 4 - Marginal Effects ————-

4.1 Average Marginal Effect for the First Model: Conditional Logit —-

```
# Coefficients
Coef_CL=res_cond_logit$par

# Compute probability matrix
x1=margarine$choicePrice[,3:12]

ut=mat.or.vec(nrow(x1),ncol(x1))
ut[,1] = Coef_CL[10]*x1[,1]
for (j in 2:ncol(x1))
{
  # conditional logit
  ut[,j] = Coef_CL[j-1] + Coef_CL[10]*x1[,j]
}
prob   = exp(ut)
prob   = sweep(prob,MARGIN=1,FUN="/",STATS=rowSums(prob))

# Computing marginal effects
avg_mgl_effects_CL <- NULL

for(j in 1:10){
mgl_effects_1=mat.or.vec(nrow(x1),ncol(x1))
dummy_reference_option <- rep(0,10)
dummy_reference_option[j] <- 1
for(jj in 1:10){
  mgl_effects_1[,jj]=prob[,jj]*(dummy_reference_option[jj]-prob[,1])*Coef_CL[10]
}
temp_avg_mgl_effects_CL=colMeans(mgl_effects_1)
avg_mgl_effects_CL <- cbind(avg_mgl_effects_CL,temp_avg_mgl_effects_CL)
}
colnames(avg_mgl_effects_CL) <- colnames(x1)
rownames(avg_mgl_effects_CL) <- colnames(x1)
avg_mgl_effects_CL
```

```
##               PPk_Stk     PBB_Stk     PFl_Stk    PHse_Stk    PGen_Stk
## PPk_Stk  -1.28534761  1.34481410  1.34481410  1.34481410  1.34481410
## PBB_Stk   0.29538486 -0.74557983  0.29538486  0.29538486  0.29538486
## PFl_Stk   0.12072430  0.12072430 -0.24117039  0.12072430  0.12072430
## PHse_Stk  0.29510268  0.29510268  0.29510268 -0.58800569  0.29510268
## PGen_Stk  0.15623726  0.15623726  0.15623726  0.15623726 -0.31286425
## PImp_Stk  0.03732068  0.03732068  0.03732068  0.03732068  0.03732068
## PSS_Tub   0.15360873  0.15360873  0.15360873  0.15360873  0.15360873
## PPk_Tub   0.09929334  0.09929334  0.09929334  0.09929334  0.09929334
## PFl_Tub   0.11083209  0.11083209  0.11083209  0.11083209  0.11083209
## PHse_Tub  0.01684364  0.01684364  0.01684364  0.01684364  0.01684364
##              PImp_Stk     PSS_Tub     PPk_Tub     PFl_Tub    PHse_Tub
## PPk_Stk   1.34481410  1.34481410  1.34481410  1.34481410  1.34481410
## PBB_Stk   0.29538486  0.29538486  0.29538486  0.29538486  0.29538486
## PFl_Stk   0.12072430  0.12072430  0.12072430  0.12072430  0.12072430
```

```
## PHse_Stk  0.29510268  0.29510268  0.29510268  0.29510268  0.29510268
## PGen_Stk  0.15623726  0.15623726  0.15623726  0.15623726  0.15623726
## PImp_Stk -0.07287453  0.03732068  0.03732068  0.03732068  0.03732068
## PSS_Tub   0.15360873 -0.32145965  0.15360873  0.15360873  0.15360873
## PPk_Tub   0.09929334  0.09929334 -0.20299249  0.09929334  0.09929334
## PFl_Tub   0.11083209  0.11083209  0.11083209 -0.22425270  0.11083209
## PHse_Tub  0.01684364  0.01684364  0.01684364  0.01684364 -0.03229780
```

Interpretation:

The average marginal effects indicate the variation in the probability of demanding option a given a change in the price of option b. If a=b, the signs are negative, as expected by demand law of regular goods. If a is different from b, the marginal effect is positive. One example would be: when the price of option 1 increases in one unit (one dollar) the probability of being demanded decreases in 1.285 percentage points.

4.2 Average Marginal Effect for the Second Model: Multinomial Logit —-

```
# Coefficients
Coef_Mult=res_multlogit$par

# Data
X <- merged_data %>% select(Income,Fam_Size,college,whtcollar,retired)
n_alternatives=10

# Compute probability matrix
ut=mat.or.vec(nrow(X),n_alternatives)
ut[,1] = 0
for (j in 2:n_alternatives){
  # multinomial logit
  ut[,j] = Coef_Mult[j-1] + Coef_Mult[j+8]*X$Income + Coef_Mult[j+17]*X$Fam_Size +
    Coef_Mult[j+26]*X$college + Coef_Mult[j+35]*X$whtcollar + Coef_Mult[j+44]*X$retired
}
prob   = exp(ut)
prob   = sweep(prob,MARGIN=1,FUN="/",STATS=rowSums(prob))
```

Computing marginal effects:

a. Income

```
mgl_effects_Income=mat.or.vec(nrow(X),n_alternatives)
Income_coefs=c(0,Coef_Mult[10:18])
wgt_avg_Income=prob%*%Income_coefs

for(j in 1:10){
  mgl_effects_Income[,j]=prob[,j]*(Income_coefs[j]-wgt_avg_Income)
}

avg_mgl_effects_Income=colMeans(mgl_effects_Income)
```

b. Fam_Size

```
mgl_effects_Fam_Size=mat.or.vec(nrow(X),n_alternatives)
Fam_Size_coefs=c(0,Coef_Mult[19:27])
wgt_avg_Fam_Size=prob%*%Fam_Size_coefs

for(j in 1:10){
  mgl_effects_Fam_Size[,j]=prob[,j]*(Fam_Size_coefs[j]-wgt_avg_Fam_Size)
}

avg_mgl_effects_Fam_Size=colMeans(mgl_effects_Fam_Size)
```

c. College

```
mgl_effects_college=mat.or.vec(nrow(X),n_alternatives)
college_coefs=c(0,Coef_Mult[28:36])
wgt_avg_college=prob%*%college_coefs

for(j in 1:10){
  mgl_effects_college[,j]=prob[,j]*(college_coefs[j]-wgt_avg_college)
}

avg_mgl_effects_college=colMeans(mgl_effects_college)
```

d. For Whtcollar

```
mgl_effects_whtcollar=mat.or.vec(nrow(X),n_alternatives)
whtcollar_coefs=c(0,Coef_Mult[37:45])
wgt_avg_whtcollar=prob%*%whtcollar_coefs

for(j in 1:10){
  mgl_effects_whtcollar[,j]=prob[,j]*(whtcollar_coefs[j]-wgt_avg_whtcollar)
}

avg_mgl_effects_whtcollar=colMeans(mgl_effects_whtcollar)
```

e. Retired

```
mgl_effects_retired=mat.or.vec(nrow(X),n_alternatives)
retired_coefs=c(0,Coef_Mult[46:54])
wgt_avg_retired=prob%*%retired_coefs

for(j in 1:10){
  mgl_effects_retired[,j]=prob[,j]*(retired_coefs[j]-wgt_avg_retired)
}

avg_mgl_effects_retired=colMeans(mgl_effects_retired)
```

Results

```r
results = cbind(avg_mgl_effects_Income,avg_mgl_effects_Fam_Size,avg_mgl_effects_college,
                avg_mgl_effects_whtcollar,avg_mgl_effects_retired)
colnames(results) = colnames(X)
rownames(results) = names(merged_data)[3:12]
results
```

```
##                 Income     Fam_Size       college     whtcollar      retired
## PPk_Stk  -1.377295e-03  0.012204564  0.0197864710 -0.0382077143 -0.031977803
## PBB_Stk  -8.396712e-04  0.003774594  0.0120462715 -0.0192239504  0.018046316
## PFl_Stk   9.737299e-04 -0.016295081  0.0325616945  0.0268051151  0.082424269
## PHse_Stk -1.759171e-04  0.024291081 -0.0186605386 -0.0097044731 -0.021534855
## PGen_Stk -7.675618e-04  0.022326874 -0.0255183892  0.0424318555  0.013544645
## PImp_Stk  4.653445e-04  0.001355287  0.0050907232 -0.0007929517  0.020366983
## PSS_Tub  -5.715764e-04 -0.015205028  0.0088046121 -0.0124443876 -0.057519631
## PPk_Tub   1.112914e-03 -0.008438946 -0.0200170185 -0.0201212884 -0.052819418
## PFl_Tub   1.232041e-03 -0.028496310 -0.0146450923  0.0148439695  0.022541331
## PHse_Tub -5.200894e-05  0.004482965  0.0005512661  0.0164138254  0.006928162
```

Interpretation:

For option 1: * An increase of one unit on income, leads to an decrease of 0.00134 percentage points on the probability of demanding option one. * An increase of one unit on Family Size, leads to an decrease of 0.0122 percentage points on the probability of demanding option one. * Those individuals who went to college, are 0.0198 percentage points more likely of demanding option one. * Those individuals with a white collar job, are 0.0382 percentage points less likely of demanding option one. * Those individuals who are retired, are 0.032 percentage points less likely of demanding option one.

——— 5 - Independence of Irrelevant Alternatives ———-

The model proposed is a mixed logit of the form:

$$Demand_{ij} = \alpha_j + \beta_j^1 * Income_i + \beta_j^2 * Fam\_Size_i + \beta_j^3 * College_i + \beta_j^4 * Whtcollar_i + \beta_j^5 * Retired_i + \beta^6 * price_{ij}$$

Unrestricted Mixed Logit

Mixed logit function

```r
Mixed_logit = function(param,choice,Product_X,Individuals_X,n_alternatives){
  #Coefficients and preliminaries
  ni = length(choice)
  nj = n_alternatives
  intercepts = c(0,param[1:(nj- 1)])
  slopes1 = c(0,param[nj:(2*(nj-1))])
  slopes2 = c(0,param[(2*(nj-1)+1):(3*(nj-1))])
  slopes3 = c(0,param[(3*(nj-1)+1):(4*(nj-1))])
  slopes4 = c(0,param[(4*(nj-1)+1):(5*(nj-1))])
```

```
    slopes5 = c(0,param[(5*(nj-1)+1):(6*(nj-1))])
    slope6 = param[length(param)]
    ut = mat.or.vec(ni,nj)

    #Variables
    Income = Individuals_X[,1]
    Fam_size = Individuals_X[,2]
    college = Individuals_X[,3]
    whtcollar = Individuals_X[,4]
    retired = Individuals_X[,5]
    price = Product_X
    # Loop to compute probabilites
    ut[,1] = slope6*price[,1] # intercept =0 and slopes=0
    for (j in 2:nj){
      # multinomial logit
      ut[,j] = intercepts[j] + slopes1[j]*Income + slopes2[j]*Fam_size +
        slopes3[j]*college + slopes4[j]*whtcollar + slopes5[j]*retired + slope6*price[,j]
    }
    prob   = exp(ut)
    prob   = sweep(prob,MARGIN=1,FUN="/",STATS=rowSums(prob))
    probc = NULL
    for (i in 1:ni){
      probc[i] = prob[i,choice[i]]
    }
    probc[probc>0.999999] = 0.999999
    probc[probc<0.000001] = 0.000001
    like = sum(log(probc))
    return(-like)
  }
```

Estimating Mixed Logit Model

```
# Dataset
Individuals_X <- merged_data %>% select(Income,Fam_Size,college,whtcollar,retired)
Product_X <- merged_data %>% select(3:12) #prices

# Initial values for model estimation
set.seed(1234)
n_alternatives=10
npar=(n_alternatives-1)*6 +1
lower  = rep(-10,npar)
upper  = rep(10,npar)
start  = runif(npar)


res_mixed=optim(start,fn=Mixed_logit,method="BFGS",control=list(trace=6,REPORT=20,maxit=10000),
                    choice=choice,Product_X=Product_X,Individuals_X=Individuals_X,n_alternatives=n_al
```

```
## initial  value 49152.075571
## iter  20 value 28992.296835
## iter  40 value 9588.372248
```

```
## iter  60 value 8076.901785
## iter  80 value 7781.777876
## final  value 7779.187888
## converged
```

```
beta_f=res_mixed$par # Coefficients
like_f=res_mixed$value # Likelihood
```

Restricted Mixed Logit

Mixed logit (removing option 10)

```
## Mixed logit removing option 10 -------
merged_data_iia <- merged_data %>% filter(choice!=10) %>% select(-PHse_Tub)

Mixed_logit_2 = function(param,choice,Product_X,Individuals_X,n_alternatives){
  #Coefficients and preliminaries
  ni = length(choice)
  nj = n_alternatives
  intercepts = c(0,param[1:(nj- 1)])
  slopes1 = c(0,param[nj:(2*(nj-1))])
  slopes2 = c(0,param[(2*(nj-1)+1):(3*(nj-1))])
  slopes3 = c(0,param[(3*(nj-1)+1):(4*(nj-1))])
  slopes4 = c(0,param[(4*(nj-1)+1):(5*(nj-1))])
  slopes5 = c(0,param[(5*(nj-1)+1):(6*(nj-1))])
  slope6 = param[length(param)]
  ut = mat.or.vec(ni,nj)

  #Variables
  Income = Individuals_X[,1]
  Fam_size = Individuals_X[,2]
  college = Individuals_X[,3]
  whtcollar = Individuals_X[,4]
  retired = Individuals_X[,5]
  price = Product_X
  # Loop to compute probabilites
  ut[,1] = slope6*price[,1] # intercept =0 and slopes=0
  for (j in 2:nj){
    # multinomial logit
    ut[,j] = intercepts[j] + slopes1[j]*Income + slopes2[j]*Fam_size +
      slopes3[j]*college + slopes4[j]*whtcollar + slopes5[j]*retired + slope6*price[,j]
  }
  prob   = exp(ut)
  prob   = sweep(prob,MARGIN=1,FUN="/",STATS=rowSums(prob))
  probc = NULL
  for (i in 1:ni){
    probc[i] = prob[i,choice[i]]
  }
  probc[probc>0.999999] = 0.999999
  probc[probc<0.000001] = 0.000001
  like = sum(log(probc))
```

```r
  return(-like)
}
```

Estimating the restricted model

```r
# Dataset
Individuals_X_iia <- merged_data_iia %>% select(Income,Fam_Size,college,whtcollar,retired)
Product_X_iia <- merged_data_iia %>% select(3:12) #prices
choice_iia <- merged_data_iia$choice

# Initial values for model estimation
set.seed(1234)
n_alternatives_iia=9
npar_iia=(n_alternatives_iia-1)*6 +1
lower  = rep(-10,npar_iia)
upper  = rep(10,npar_iia)
start  = runif(npar_iia)



res_mixed_iia=optim(start,fn=Mixed_logit_2,method="BFGS",control=list(trace=6,REPORT=20,maxit=10000),
                choice=choice_iia,Product_X=Product_X_iia,Individuals_X=Individuals_X_iia,
                n_alternatives=n_alternatives_iia,hessian=FALSE)
```

```
## initial  value 42368.532207
## iter  20 value 18265.317865
## iter  40 value 12107.028430
## iter  60 value 9338.858636
## iter  80 value 8502.547358
## iter 100 value 8151.052434
## iter 120 value 8115.627965
## final  value 8115.572095
## converged
```

```r
beta_r=res_mixed_iia$par # Coefficients
like_r=res_mixed_iia$value # Likelihood
```

Testing for IIA

To compute the MTT statistics we have to compute the likelihood of the restricted model with the coefficients of the free model and the restricted model

```r
## For the Lr using the unrestricted coefficients: Lr_beta_f

# Observations are the ones from the restricted model
Individuals_X_iia <- merged_data_iia %>% select(Income,Fam_Size,college,whtcollar,retired)
Product_X_iia <- merged_data_iia %>% select(3:12) #prices
choice_iia <- merged_data_iia$choice
```

```r
# Loop to compute probabilites
# Coefficiens
intercepts = beta_f[1:8]
slopes1 = beta_f[10:17]
slopes2 = beta_f[19:26]
slopes3 = beta_f[28:35]
slopes4 = beta_f[37:44]
slopes5 = beta_f[46:53]
slope6 = beta_f[length(beta_f)]
param_beta_f_for_iia=c(intercepts,slopes1,slopes2,slopes3,slopes4,slopes5,slope6)

#Variables
Income = Individuals_X_iia[,1]
Fam_size = Individuals_X_iia[,2]
college = Individuals_X_iia[,3]
whtcollar = Individuals_X_iia[,4]
retired = Individuals_X_iia[,5]
price = Product_X_iia

# Loop to compute probabilites
ni = length(choice_iia)
nj = 9
ut = mat.or.vec(ni,nj)
ut[,1] = slope6*price[,1] # intercept =0 and slopes=0
for (j in 2:nj){
  # mixed logit
  ut[,j] = intercepts[j-1] + slopes1[j-1]*Income + slopes2[j-1]*Fam_size +
    slopes3[j-1]*college + slopes4[j-1]*whtcollar + slopes5[j-1]*retired + slope6*price[,j-1]
}
prob   = exp(ut)
prob   = sweep(prob,MARGIN=1,FUN="/",STATS=rowSums(prob))
probc = NULL
for (i in 1:ni){
  probc[i] = prob[i,choice_iia[i]]
}
probc[probc>0.999999] = 0.999999
probc[probc<0.000001] = 0.000001
Lr_beta_f = sum(log(probc))
```

Computing the MTT statistic and testig

```r
mtt=-2*(Lr_beta_f-(-like_r))
chi_95=qchisq(0.95,df=length(beta_r))
mtt>chi_95
```

```
## [1] TRUE
```

Interpretation:

The fact that the MTT statistic is larger than the

$$\chi^2$$

leads to rejecting the null hypothesis that IIA is not violated. Hence we can not conclude that IIA holds.